

# Comparison of Different Cue-Based Swarm Aggregation Strategies

Farshad Arvin<sup>1</sup>, Ali Emre Turgut<sup>2</sup>, Nicola Bellotto<sup>1</sup>, and Shigang Yue<sup>1</sup>

<sup>1</sup> Computational Intelligence Lab (CIL), School of Computer Science, University of Lincoln, Lincoln, LN6 7TS, UK

<sup>2</sup> Laboratory of Socioecology and Social Evolution, KU Leuven, B-3000 Leuven, Belgium  
{farvin, syue}@lincoln.ac.uk

**Abstract.** In this paper, we compare different aggregation strategies for cue-based aggregation with a mobile robot swarm. We used a sound source as the cue in the environment and performed real robot and simulation based experiments. We compared the performance of two proposed aggregation algorithms we called as the *vector averaging* and *naïve* with the state-of-the-art cue-based aggregation strategy BEECLUST. We showed that the proposed strategies outperform BEECLUST method. We also illustrated the feasibility of the method in the presence of noise. The results showed that the vector averaging algorithm is more robust to noise when compared to the naïve method.

**Keywords:** swarm robotics, collective behavior, cue-based aggregation.

## 1 Introduction

Aggregation is a widely observed phenomenon in social insects and animals such as cockroaches, honeybees and birds [1]. It provides additional capabilities to animals such as forming a spore-bearing structure by slime mold or building a nest by termites [2]. In general, two types of aggregation mechanisms are observed in nature: cue-based or clueless. In cue-based aggregation, animals follow external cues to identify optimal zones such as a humid location for sow bugs and then they aggregate on these zones. Whereas in clueless aggregation, animals aggregate at random locations in an environment such as aggregation of cockroaches [3]. From swarm robotics perspective [4], aggregation can be defined as gathering randomly distributed robots to form an aggregate. Due to limited sensing capabilities of the robots, aggregation turns out to be one of the challenging tasks in swarm robotics.

Many different studies have been performed in cue-based and clueless aggregation in swarm robotics. We first discuss cue-based aggregation. In one of the earliest studies on cue-based aggregation, Kube and Zhang [5] proposed an aggregation algorithm in which robots are required to aggregate around a light box and then push it. Melhuish et al. [6] proposed an algorithm for aggregation of robots around an infrared (IR) transmitter. The robots after reaching the IR transmitter start to emit sound resembling the vocalization of frogs and birds in order to help the other robots to estimate the size of the aggregate they are in. Honeybee aggregation is another example of cue-based aggregation method that was studied in [7,8]. In these studies, micro robots were deployed in a gradually lighted environment to mimic the behavior of honeybees which

aggregate at a zone that has the optimal temperature. An aggregation algorithm called BEECLUST that relies on inter-robot collisions was proposed [9]. The aggregation method has been used and evaluated in several researches [10,11,12]. In another study [13], two modifications on BEECLUST – *dynamic velocity* and *comparative waiting time* – were applied to increase the performance of aggregation. In addition, a fuzzy-based reasoning method has been proposed in [14,15] which increases the performance of the system significantly.

Clueless aggregation mechanism was employed in various studies. Trianni et al. [16] proposed an aggregation behavior of mobile robots using artificial evolution with static and dynamic strategies. In static strategy, when robots create an aggregate, they are not allowed to leave it. However, in the dynamic strategy, robots are allowed to leave an aggregate with a certain probability depending on the size of the aggregate. Static strategy resulted in many small compact aggregates, whereas with the dynamic one robots are able to form a few larger aggregates. Soysal and Şahin [17] proposed a probabilistic aggregation method using a combination of basic behaviors: obstacle avoidance, approach, repel, and wait. They studied the effects of various parameters such as control mechanisms, time, and arena size in performance of the system. This study was continued by using an evolutionary approach in order to investigate the various effective parameters in aggregation, such as the number of generations, the number of simulation steps which are used for fitness evaluations, population size, and size of arena [18]. Bayrindir and Şahin [19] proposed a macroscopic model to study the effects of population size and probability of leaving an aggregate on aggregation performance in a clueless aggregation scenario.

In this paper, different than the previous studies, we compare three different aggregation mechanisms in a cue-based aggregation scenario. Specifically, we extend the BEECLUST algorithm and propose two new algorithms which we call as *vector averaging* and *naïve* aggregation algorithms. Through systematic real-robot and simulation-based experiments, we analyze and compare the performance differences between those two algorithms and the BEECLUST algorithm.

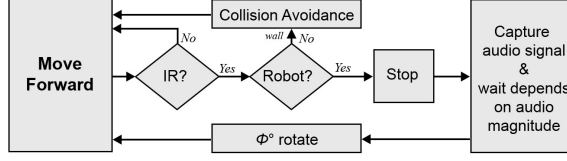
## 2 Aggregation Methods

### 2.1 BEECLUST

BEECLUST aggregation [9] follows a simple algorithm as shown in the Fig. 1. When a robot detects another robot in the environment, it stops and measures the magnitude of the ambient audio signal and waits based on this magnitude. The higher the magnitude is the longer the waiting time ( $w$ ) becomes. The waiting time is estimated by the following formula assuming that four microphones are used to detect the ambient audio signal:

$$w(t) = w_{max} \cdot \frac{M_a(t)^2}{M_a(t)^2 + \mu}, \quad (1)$$

where  $M_a(t) = \frac{1}{4} \sum_{i=1}^4 M_i(t)$  is the average magnitude of the four microphones,  $M_i$  is the magnitude of signal from the  $i$ th microphone ranging from 0 and 255,  $w_{max}$  is the maximum waiting time, and  $\mu$  is a parameter which changes the steepness of the waiting curve.  $w_{max}$  and  $\mu$  are determined empirically:  $w_{max} = 65$  sec and  $\mu = 5500$ .



**Fig. 1.** Control diagram of the aggregation

When the waiting time is over, the robot rotates  $\phi$  degree, which is a random variable drawn uniformly within  $[-180^\circ, 180^\circ]$ .

## 2.2 Naïve Method

In the naïve aggregation method, we employ a deterministic decision making mechanism based on both the intensity and the direction of the sound signal. The waiting time is still calculated using (1) based on the average intensity ( $M_a$ ), but in addition to that, we estimate the direction of the sound source by setting it to the angle of the sound sensor that has the highest reading.  $\phi = \theta_i$ ,  $i \in \{1, 2, 3, 4\}$  and  $\theta_i$  is the angle of the sensor  $\{45^\circ, 135^\circ, 225^\circ, 315^\circ\}$  with respect to the frontal axis of the robot having the highest reading.

## 2.3 Vector Averaging Method

In vector averaging method, the direction information and intensity of the sound source are utilized. We employ an averaging calculation based on both intensity and direction of sound signal to estimate  $\phi$ .

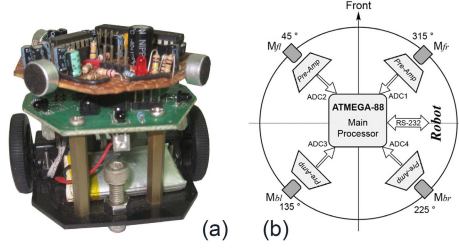
$$\phi = \text{atan2} \left( \frac{\sum_{i=1}^4 \hat{M}_i \sin(\beta_i)}{\sum_{i=1}^4 \hat{M}_i \cos(\beta_i)} \right) \quad (2)$$

$\phi$  is the estimated angular position of the source speaker,  $\beta_i$  is the angular distance between  $i$ th microphone and the robot's head.  $\hat{M}_i$ ,  $i \in \{1, 2, 3, 4\}$  is the captured audio signal's intensity levels from microphone  $i$ .

# 3 Experimental Setup

## 3.1 Robot Platform

AMiR (Autonomous Miniature Robot) as shown in Fig. 2 (a) is an open hardware mobile robot platform [20]. AMiR is specifically developed for swarm robotics studies. The robot has two small geared DC motors that make it move with a maximum speed of  $8.6 \frac{\text{cm}}{\text{s}}$  [21]. Six IR proximity sensors using  $60^\circ$  topology is used, which allows AMiR to scan its surrounding area without turning [22]. AMiR also has an audio extension module which is composed of four condenser microphones ( $45^\circ, 135^\circ, 225^\circ$ , and  $315^\circ$ ) as shown in Fig. 2 (b).



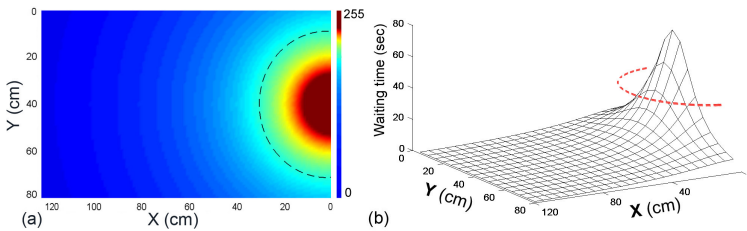
**Fig. 2.** (a) Autonomous Miniature Robot (AMiR) is equipped with an audio extension module and (b) architecture of developed audio signal processor module

### 3.2 Simulation Software

In order to test the proposed algorithm in large scale, Player/Stage simulation software is utilized. We modeled AMiR and its sensors in Player and used Stage as the simulation platform.

### 3.3 Experiment Configuration

**Real Robot Experiments.** In real robot experiments, due to laboratory limitations, a rectangular arena with a size of  $120 \times 80 \text{ cm}^2$  is utilized. When compared to the sensing range of the robots, the arena is approximately two times larger than the total sensing area of 6 robots. The experiments are performed with different number of robots  $N \in \{3, 4, 5, 6\}$ . We placed a sound source at one of the edges of the arena, that plays a single tone of frequency of approximately 1050 Hz. The sound source serves as the environmental cue in the experiment. With the current configuration of the arena, the intensity and waiting times are shown in Fig. 3. Each experiment is repeated 10 times and at the start of each run, the robots are placed in the arena with random positions and orientations.



**Fig. 3.** (a) Intensity of audio signals in the arena and (b) relative waiting time in different positions of the arena. Dashed arc shows the predefined optimal aggregation zone around the sound source.

**Simulation-Based Experiments.** In simulation-based experiments, in order to use large number of robots  $N \in \{5, 10, 15, 20, 25\}$ , we used an arena with a size of  $240 \times 160 \text{ cm}^2$ . Similar to the real-robot experiments, the experiments are repeated 10 times and position and orientation of the robots are set randomly at the beginning of each experiment. We perform two sets of experiments: One set without sensing noise, the other with sensing noise. In the latter, we add noise to the sound measurements, which is modeled as a uniformly distributed random variable. For the  $i$ th microphone, the noise is added as  $\overline{M}_i = |M_i + \sigma\rho|$  where  $M_i$  is sensor reading of  $i$ th microphone,  $\sigma \in \{0.1, 0.3, 0.5\}$  is the noise scaling factor which determines the amount of noise to be added to the reading and  $\rho$  is a random value within  $[-255, +255]$ .

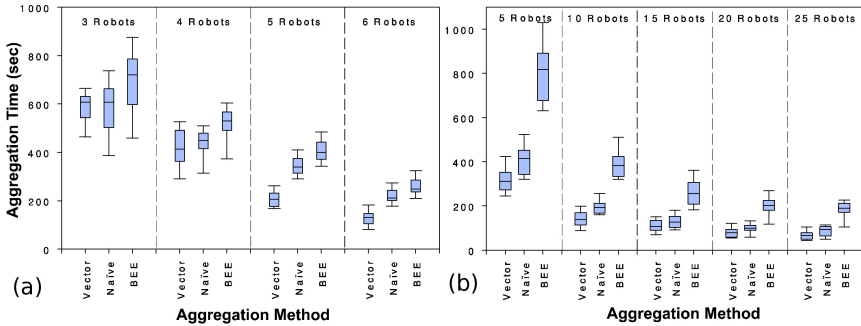
### 3.4 Metrics

In this study, we are interested in having fast aggregation around the sound source. Therefore, we use aggregation time as one of our metrics. In order to calculate aggregation time,  $T$ , we define an aggregation area which is shown with a dashed arc in Fig. 3. The aggregation time is the duration of an experiment in which the number of the robots aggregated in the aggregation area reaches 80% of the total robots.

## 4 Results and Discussion

### 4.1 Real Robot Experiments

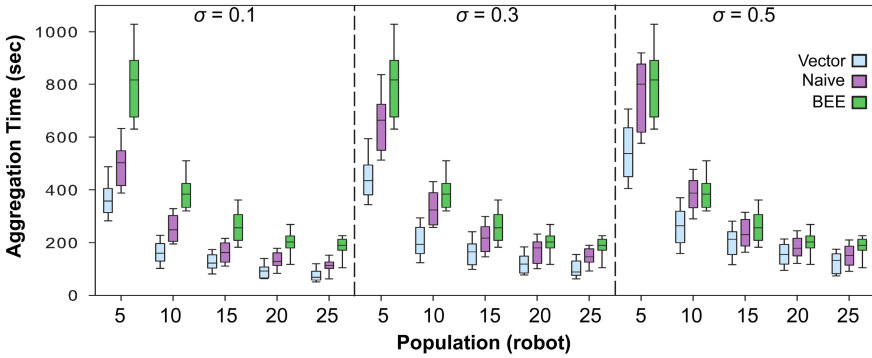
We first performed experiments with real robots. In general, when the number of robots increases, the aggregation time reduces significantly as shown in Fig. 4a, since increasing the population size increases the number of collisions eventually causing faster aggregation. It should be noted that, the reduction rate in the aggregation time is not the same in the three algorithms. Results show that, vector averaging aggregation is faster than naïve and BEECLUST.



**Fig. 4.** Aggregation time as a function of population size for vector averaging, naïve, and BEECLUST aggregation methods with (a) real robot and (b) simulated robot experiments

## 4.2 Simulation-Based Experiments

In these experiments, the aggregation methods are implemented using the simulator. Fig. 4b shows the aggregation time as a function of population size in vector averaging, naïve, and BEECLUST methods. Results show that, population size has a direct impact on the aggregation time. Simulation results also showed that the vector averaging method performs faster aggregation in comparison with naïve and BEECLUST owing to more precise estimates  $\phi$  values after each collision, which increases the performance of the aggregation. In case of BEECLUST that relies on random rotation, the aggregation time is longer than the others. Due to random rotations, robots occasionally move in opposite direction of the sound source, that results in robots leaving the aggregation zone. Therefore, it increases the aggregation time and number of collisions.



**Fig. 5.** Effects of different noise values in different population sizes for vector averaging, naïve and BEECLUST methods

Although aggregation time decreases with increasing population size, when the density of swarm reaches a certain value the performance decreases. This is due to over-crowding effect observed, when the density of robots reaches a certain value [23]. In our current setting, the performance increases up to 25 robots.

The results of the experiments performed with sensing noise is depicted in Fig. 5. The rotation angle,  $\phi$ , for naïve and vector averaging methods relies on the prediction of the direction of the sound source. However, in BEECLUST method which a random value is used as the rotation angle, robots do not employ the magnitude of the audio signal. Hence, only naïve and vector averaging methods are tested against sensing noise. Results show that the vector averaging algorithm is more robust to noise than naïve algorithm.

## 4.3 Statistical Analysis

We statistically analyzed the results of simulated robot experiments with and without noise using analysis of variance (ANOVA), the F-test method (see Table 1).

Population size has the highest influence on aggregation time for the vector averaging method ( $F_p = 116.60$ ), which means that increasing the swarm density improves the performance of the vector averaging method the most. Noise has the least influence on the vector averaging method (7.327, and 12.39 for the vector averaging and naïve methods, respectively).

**Table 1.** Results of F-test in analysis of variance (ANOVA)

Experimental Setup	Parameters	Aggregation Method		
		<i>Averaging</i>	<i>Naïve</i>	<i>BEECLUST</i>
Without Noise	Population	116.60	85.51	76.75
With Noise	Noise	7.32	12.39	–
	Population	34.28	28.43	–

## 5 Conclusion

In this paper we evaluate three cue-based aggregation methods, namely, vector averaging, naïve and BEECLUST. The aggregation methods were implemented using real- and simulated-robots with different population sizes. The results showed that the proposed vector averaging method improves the performance of the aggregation significantly. The swarm density has direct impact on the performance of the aggregation. Hence, an increase in population improves the performance of aggregation in all three algorithms. In addition to these, results revealed that the additional noise has less impact on vector averaging method in comparison to the naïve method.

**Acknowledgments.** This work was supported by EU FP7-IRSES projects EYE2E (269118), LIVCODE (295151) and HAZCEPT (318907). The second author would thank TUBITAK-2219-Program.

## References

1. Camazine, S., Franks, N., Sneyd, J., Bonabeau, E., Deneubourg, J.L., Theraulaz, G.: Self-organization in biological systems. Princeton University Press (2003)
2. Parrish, J., Edelstein-Keshet, L.: Complexity, pattern, and evolutionary trade-offs in animal aggregation. *Science* 284(5411), 99–101 (1999)
3. Jeanson, R., Rivault, C., Deneubourg, J.L., Blanco, S., Fournier, R., Jost, C., Theraulaz, G.: Self-organized aggregation in cockroaches. *Animal Behaviour* 69(1), 169–180 (2005)
4. Şahin, E., Girgin, S., Bayındır, L., Turgut, A.E.: Swarm robotics. In: Blum, C., Merkle, D. (eds.) *Swarm Intelligence*, vol. 1, pp. 87–100. Springer, Heidelberg (2008)
5. Kube, C., Zhang, H.: Collective robotics: From social insects to robots. *Adaptive Behavior* 2(2), 189–219 (1993)
6. Melhuish, C., Holland, O., Hoddell, S.: Convoying: Using chorusing to form travelling groups of minimal agents. *Robotics and Autonomous Systems* 28(2), 207–216 (1999)
7. Schmickl, T., Thenius, R., Moeslinger, C., Radspieler, G., Kernbach, S., Szymanski, M., Crailsheim, K.: Get in touch: Cooperative decision making based on robot-to-robot collisions. *Autonomous Agents and Multi-Agent Systems* 18(1), 133–155 (2009)

8. Kernbach, S., Thenius, R., Kernbach, O., Schmickl, T.: Re-embodiment of Honeybee Aggregation Behavior in an Artificial Micro-Robotic System. *Adaptive Behavior* 17(3), 237–259 (2009)
9. Schmickl, T., Hamann, H.: Beeclust: A swarm algorithm derived from honeybees. In: Xiao, Y., Hu, F. (eds.) *Bio-inspired Computing and Communication Networks* (2010)
10. Bodi, M., Thenius, R., Szopek, M., Schmickl, T., Crailsheim, K.: Interaction of robot swarms using the honeybee-inspired control algorithm beeclust. *Mathematical and Computer Modelling of Dynamical Systems* 18(1), 87–100 (2012)
11. Kengyel, D., Thenius, R., Crailsheim, K., Schmickl, T.: Influence of a social gradient on a swarm of agents controlled by the beeclust algorithm. In: *European Conference on Artificial Life*, pp. 1041–1048 (2013)
12. Hereford, J.: Beeclust swarm algorithm: Analysis and implementation using a markov chain model. *International Journal of Innovative Computing and Applications* 5(2), 115–124 (2013)
13. Arvin, F., Samsudin, K., Ramli, A.R., Bekravi, M.: Imitation of honeybee aggregation with collective behavior of swarm robots. *International Journal of Computational Intelligence Systems* 4(4), 739–748 (2011)
14. Arvin, F., Turgut, A.E., Yue, S.: Fuzzy-based aggregation with a mobile robot swarm. In: Dorigo, M., Birattari, M., Blum, C., Christensen, A.L., Engelbrecht, A.P., Groß, R., Stützle, T. (eds.) *ANTS 2012. LNCS*, vol. 7461, pp. 346–347. Springer, Heidelberg (2012)
15. Arvin, F., Turgut, A.E., Bazyari, F., Arıkan, K.B., Bellotto, N., Yue, S.: Cue-based aggregation with a mobile robot swarm: A novel fuzzy-based method. *Adaptive Behavior* 22(3), 189–206 (2014)
16. Trianni, V., Groß, R., Labella, T.H., Şahin, E., Dorigo, M.: Evolving aggregation behaviors in a swarm of robots. In: Banzhaf, W., Ziegler, J., Christaller, T., Dittrich, P., Kim, J.T. (eds.) *ECAL 2003. LNCS (LNAI)*, vol. 2801, pp. 865–874. Springer, Heidelberg (2003)
17. Soysal, O., Şahin, E.: Probabilistic aggregation strategies in swarm robotic systems. In: *Swarm Intelligence Symposium*, pp. 325–332. IEEE (2005)
18. Soysal, O., Bahçeci, E., Şahin, E.: Aggregation in swarm robotic systems: Evolution and probabilistic control. *Turkish Journal of Electrical Engineering & Computer Sciences* 15(2), 199–225 (2007)
19. Bayındır, L., Şahin, E.: Modeling self-organized aggregation in swarm robotic systems. In: *Swarm Intelligence Symposium*, pp. 88–95. IEEE (2009)
20. Arvin, F., Samsudin, K., Ramli, A.R.: Development of a Miniature Robot for Swarm Robotic Application. *International Journal of Computer and Electrical Engineering* 1, 436–442 (2009)
21. Arvin, F., Bekravi, M.: Encoderless position estimation and error correction techniques for miniature mobile robots. *Turkish Journal of Electrical Engineering & Computer Sciences* 21, 1631–1645 (2013)
22. Arvin, F., Samsudin, K., Ramli, A.R.: Development of IR-Based Short-Range Communication Techniques for Swarm Robot Applications. *Advances in Electrical and Computer Engineering* 10(4), 61–68 (2010)
23. Hamann, H.: Towards swarm calculus: Universal properties of swarm performance and collective decisions. In: Dorigo, M., Birattari, M., Blum, C., Christensen, A.L., Engelbrecht, A.P., Groß, R., Stützle, T. (eds.) *ANTS 2012. LNCS*, vol. 7461, pp. 168–179. Springer, Heidelberg (2012)